Communication Cost in Big Data Processing

Dan Suciu
University of Washington

Joint work with Paul Beame and Paris Koutris and the Database Group at the UW
Queries on Big Data

Big Data processing on distributed clusters
• General programs: MapReduce, Spark
• SQL: PigLatin,, BigQuery, Shark, Myria…

Traditional Data Processing
• Main cost = disk I/O

Big Data Processing
• Main cost = communication
This Talk

• How much communication is needed to solve a “problem” on $p$ servers?

• “Problem”:
  – In this talk = Full Conjunctive Query CQ ($\approx$ SQL)
  – Future work = extend to linear algebra, ML, etc.

• Some techniques discussed in this talk are implemented in Myria (Bill Howe’s talk)
Models of Communication

• Combined communication/computation cost:
  – PRAM $\rightarrow$ MRC [Karloff’2010]
  – BSP [Valiant] $\rightarrow$ LogP Model [Culler]

• Explicit communication cost
  – [Hong&Kung’81] red/blue pebble games, for I/O communication complexity $\rightarrow$
    [Ballard’2012] extension to parallel algorithms
  – MapReduce model [DasSarma’2013] MPC [Beame’2013,Beame’2014] – this talk
Outline

• The MPC Model

• Communication Cost for Triangles

• Communication Cost for CQ’s

• Discussion
Massively Parallel Communication Model (MPC)

Extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

Input (size = $M$)

Server 1

$O(M/p)$

Server $p$

$O(M/p)$
Massively Parallel Communication Model (MPC)

Extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Input (size=$M$)

$O(M/p)$

Round 1
Massively Parallel Communication Model (MPC)

Extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Algorithm = Several rounds
Massively Parallel Communication Model (MPC)

Extends BSP [Valiant]

Input data = size $M$ bits

Number of servers = $p$

One round = Compute & communicate

Algorithm = Several rounds

Max communication load per server = $L$

Input (size = $M$) $\rightarrow$ O($M/p$)

Server 1 $\rightarrow$ Server p

Round 1

Round 2

Round 3

$\ldots$

$\ldots$

$O(M/p)$

$O(M/p)$
Extends BSP [Valiant]

Input data = size \( M \) bits

Number of servers = \( p \)

One round = Compute & communicate

Algorithm = Several rounds

Max communication load per server = \( L \)

Ideal: \( L = \frac{M}{p} \)

This talk: \( L = \frac{M}{p} \cdot p^\epsilon \), where \( \epsilon \) in \([0,1]\) is called space exponent

Degenerate: \( L = M \) (send everything to one server)
Example: $Q(x,y,z) = R(x,y) \bowtie S(y,z)$

**Input:**
- $R$, $S$ uniformly partitioned on $p$ servers

$\text{size}(R) + \text{size}(S) = M$

$\frac{M}{p}$
Example: $Q(x,y,z) = R(x,y) \bowtie S(y,z)$

**Input:**
- $R$, $S$ uniformly partitioned on $p$ servers

**Round 1:** each server
- Sends record $R(x,y)$ to server $h(y) \mod p$
- Sends record $S(y,z)$ to server $h(y) \mod p$

Input:
- $R$, $S$ uniformly partitioned on $p$ servers

Round 1: each server
- Sends record $R(x,y)$ to server $h(y) \mod p$
- Sends record $S(y,z)$ to server $h(y) \mod p$
Example: \( Q(x,y,z) = R(x,y) \bowtie S(y,z) \)

**Input:**
- \( R, S \) uniformly partitioned on \( p \) servers

**Round 1:** each server
- Sends record \( R(x,y) \) to server \( h(y) \mod p \)
- Sends record \( S(y,z) \) to server \( h(y) \mod p \)

**Output:**
- Each server computes and outputs the local join \( R(x,y) \bowtie S(y,z) \)
Example: \( Q(x,y,z) = R(x,y) \bowtie S(y,z) \)

**Input:**
• \( R, S \) uniformly partitioned on \( p \) servers

**Round 1:** each server
• Sends record \( R(x,y) \) to server \( h(y) \mod p \)
• Sends record \( S(y,z) \) to server \( h(y) \mod p \)

**Output:**
• Each server computes and outputs the local join \( R(x,y) \bowtie S(y,z) \)

Assuming no Skew: \( \forall y, \) \( \deg R(y), \deg S(y) \leq \frac{M}{p} \)

Rounds = 1
Load \( L = O(\frac{M}{p}) = O(\frac{M}{p} \cdot p^0) \)
Example: $Q(x,y,z) = R(x,y) \bowtie S(y,z)$

Input:
- $R$, $S$ uniformly partitioned on $p$ servers

Round 1:
- Each server sends $R(x,y)$ to server $h(y) \mod p$
- Each server sends $S(y,z)$ to server $h(y) \mod p$

Output:
- Each server computes and outputs the local join $R(x,y) \bowtie S(y,z)$

Assuming no Skew: $\forall y$, degree$_R(y)$, degree$_S(y) \leq M/p$

Rounds = 1
Load $L = O(M/p) = O(M/p \times p^0)$

SQL is embarrassingly parallel!
Questions

Fix a query $Q$ and a number of servers $p$

• What is the communication load $L$ needed to compute $Q$ in one round? This talk based on [Beame’2013, Beame’2014]

• Fix a maximum load $L$ (space exponent $\varepsilon$) How many rounds are needed for $Q$? Preliminary results in [Beame’2013]
Outline

• The MPC Model

• Communication Cost for Triangles

• Communication Cost for CQ’s

• Discussion
The Triangle Query

- **Input**: three tables
  \[ R(X, Y), \ S(Y, Z), \ T(Z, X) \]
  \[ \text{size}(R) + \text{size}(S) + \text{size}(T) = M \]

- **Output**: compute
  \[ Q(x,y,z) = R(x,y) \bowtie S(y,z) \bowtie T(z,x) \]
Triangles in Two Rounds

• Round 1: compute
  \[\text{temp}(X,Y,Z) = R(X,Y) \bowtie S(Y,Z)\]

• Round 2: compute
  \[Q(X,Y,Z) = \text{temp}(X,Y,Z) \bowtie T(Z,X)\]

Load \(L\) depends on the size of \text{temp}. Can be much larger than \(M/p\)
Triangles in One Round

[Ganguli’92, Afrati&Ullman’10, Suri’11, Beame’13]

• Factorize $p = p^{1/3} \times p^{1/3} \times p^{1/3}$
• Each server identified by $(i,j,k)$

Place the servers in a cube:

\[Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)\]

size($R$)+size($S$)+size($T$)=M
Triangles in One Round

Round 1:
- Send $R(x,y)$ to all servers $(h(x), h(y), \ast)$
- Send $S(y,z)$ to all servers $(\ast, h(y), h(z))$
- Send $T(z,x)$ to all servers $(h(x), \ast, h(z))$

Output:
compute locally $R(x,y) \bowtie S(y,z) \bowtie T(z,x)$

\[ Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \]

\[
\text{size}(R) + \text{size}(S) + \text{size}(T) = M
\]
Theorem Assume data has no skew: all degree \( \leq O(M/p^{1/3}) \). Then the algorithm computes \( Q \) in one round, with communication load \( L_{\text{algo}} = O(M/p^{2/3}) \).

Compare to two rounds:
- No more large intermediate result
- Skew-free up to degrees = \( O(M/p^{1/3}) \) (from \( O(M/p) \))
- BUT: space exponent increased to \( \varepsilon = \frac{1}{3} \) (from \( \varepsilon = 1 \))
Lower Bound is $L_{\text{lower}} = \frac{M}{p^{2/3}}$
Lower Bound is $L_{\text{lower}} = \frac{M}{p^{2/3}}$

$M = \text{size}(R) + \text{size}(S) + \text{size}(T)$ (in bits)

Assume that the three input relations $R$, $S$, $T$ are stored on disjoint servers

**Theorem** *Any one-round deterministic algorithm $A$ for $Q$ requires $L_{\text{lower}}$ bits of communication per server*
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)

size(R)+size(S)+size(T)=M

Lower Bound is \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

\( M = \text{size}(R)+\text{size}(S)+\text{size}(T) \) (in bits)

Assume that the three input relations R, S, T are stored on disjoint servers#

**Theorem** * Any one-round deterministic algorithm A for Q requires \( L_{\text{lower}} \) bits of communication per server

We prove a stronger claim, about any algorithm communicating < \( L_{\text{lower}} \) bits

Let \( L \) be the algorithm’s max communication load per server

**We prove**: if \( L < L_{\text{lower}} \) then, over random permutations R, S, T, the algorithm returns at most a fraction \( \left( \frac{L}{L_{\text{lower}}} \right)^{3/2} \) of the answers to Q

---

# without this assumption need to add a multiplicative factor 3

* Beame, Koutris, Suciu, Communication Steps for Parallel Query Processing, PODS 2013
Lower Bound is \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

Discussion:

• An algorithm with space exponent \( \varepsilon < \frac{1}{3} \), has load \( L = \frac{M}{p^{1-\varepsilon}} \) and reports only \( \left( \frac{L}{L_{\text{lower}}} \right)^{3/2} = \frac{1}{p^{(1-3\varepsilon)/2}} \) fraction of all answers. Fewer, as \( p \) increases!

• Lower bound holds only for random inputs: cannot hold for fixed input \( R, S, T \) since the algorithm may use a short bit encoding to signal this input to all servers.

• By Yao’s lemma, the result also holds for randomized algorithms, some fixed input
Lower Bound is \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

Balard’2012] consider Strassen’s matrix multiplication algorithm and prove the following theorem for the CDAG model. If each server has memory, then the communication load per server:

\[
\text{IO}(n, p, M) = \Omega \left( \left( \frac{n}{\sqrt{M}} \right)^{\lg 7} \frac{M}{p} \right)
\]

Thus, if \( M = \frac{n^2}{p^{2/\lg 7}} \) then \( \text{IO}(n, p) = \Omega \left( \frac{n^2}{p^{2/\lg 7}} \right) \)

- Stronger than MPC: no restriction on rounds
- Weaker than MPC
  - Applies only to an algorithm, not to a problem
  - CDAG model of computation v.s. bit-model

\[
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)
\]

\[
\text{size}(R) + \text{size}(S) + \text{size}(T) = M
\]
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)

Lower Bound is $L_{\text{lower}} = \frac{M}{p^{2/3}}$

Proof
\[ Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \]

Lower Bound is \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

Proof

Input: size = \( M = 3 \log n \)!

- \( R, S, T \) random permutations on \([n]\): \( \Pr [(i,j) \in R] = 1/n\), same for \( S, T \)

\[
E [\#\text{answers to } Q] = \sum_{i,j,k} \Pr [(i,j,k) \in R \bowtie S \bowtie T] = n^3 \times (1/n)^3 = 1
\]
\[
Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)
\]

Lower Bound is \( L_{\text{lower}} = \frac{M}{p^{2/3}} \)

Proof

Input: size = \( M = 3 \log n! \)
- \( R, S, T \) random permutations on \([n]\): \( \Pr [(i,j) \in R] = 1/n \), same for \( S, T \)

\[
\mathbb{E} [\#\text{answers to } Q] = \sum_{i,j,k} \Pr [(i,j,k) \in R \bowtie S \bowtie T] = n^3 \times (1/n)^3 = 1
\]

One server \( v \in [p] \): receives \( L(v) = L_R(v) + L_S(v) + L_T(v) \) bits
- Denote \( a_{ij} = \Pr [(i,j) \in R \text{ and } v \text{ knows it}] \)
  Then (1) \( a_{ij} \leq 1/n \) (obvious) (2) \( \sum_{i,j} a_{ij} \leq L_R(v) / \log n \) (formal proof: entropy)
- Denote similarly \( b_{jk}, c_{ki} \) for \( S \) and \( T \)
Lower Bound is $L_{\text{lower}} = M / p^{2/3}$

Proof

**Input:** size = $M = 3 \log n$!

- $R$, $S$, $T$ random permutations on $[n]$: $\Pr [(i,j) \in R] = 1/n$, same for $S$, $T$

**One server** $v \in [p]$ receives $L(v) = L_R(v) + L_S(v) + L_T(v)$ bits

- Denote $a_{ij} = \Pr [(i,j) \in R \text{ and } v \text{ knows it}]$
  Then (1) $a_{ij} \leq 1/n$ (obvious) (2) $\sum_{i,j} a_{ij} \leq L_R(v) / \log n$ (formal proof: entropy)

- Denote similarly $b_{jk}$, $c_{ki}$ for $S$ and $T$

**$E$ [#answers to $Q$ reported by server $v$] =**

$$
\sum_{i,j,k} a_{ij} b_{jk} c_{ki} \leq \left(\sum_{i,j} a_{ij}^2\right) \cdot \left(\sum_{j,k} b_{jk}^2\right) \cdot \left(\sum_{k,i} c_{ki}^2\right)^{1/2} \quad \text{[Friedgut]}
$$

$$
\leq (1/n)^{3/2} \cdot \left[ L_R(v) \cdot L_S(v) \cdot L_T(v) / \log^3 n \right]^{1/2} \quad \text{by (1), (2)}
$$

$$
\leq (1/n)^{3/2} \cdot \left[ L(v) / 3 \cdot \log n \right]^{3/2}
$$

$$
= \left[ L(v) / M \right]^{3/2}
$$

$Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X)$

$\text{size}(R)+\text{size}(S)+\text{size}(T)=M$
Lower Bound is \( L_{\text{lower}} = M / p^{2/3} \)

**Proof**

**Input:** size = \( M = 3 \log n \)

- \( R, S, T \) random permutations on \([n] \): \( \Pr [(i,j) \in R] = 1/n \), same for \( S, T \)

\[
E \left[ \#\text{answers to } Q \right] = \sum_{i,j,k} \Pr [(i,j,k) \in R \bowtie S \bowtie T] = n^3 \times (1/n)^3 = 1
\]

One server \( v \in [p] \): receives \( L(v) = L_R(v) + L_S(v) + L_T(v) \) bits

- Denote \( a_{ij} = \Pr [(i,j) \in R \text{ and } v \text{ knows it}] \)
  Then \( (1) a_{ij} \leq 1/n \) (obvious) \( (2) \sum_{i,j} a_{ij} \leq L_R(v) / \log n \) (formal proof: entropy)

- Denote similarly \( b_{jk}, c_{ki} \) for \( S \) and \( T \)

\[
E \left[ \#\text{answers to } Q \text{ reported by server } v \right] = \sum_{i,j,k} a_{ij} b_{jk} c_{ki} \leq [\left( \sum_{i,j} a_{ij}^2 \right) \times \left( \sum_{j,k} b_{jk}^2 \right) \times \left( \sum_{k,i} c_{ki}^2 \right)]^{1/2} \quad \text{[Friedgut]}
= (1/n)^{3/2} \times \left[ L_R(v) \times L_S(v) \times L_T(v) / \log^3 n \right]^{1/2} \quad \text{by (1), (2)}
\leq (1/n)^{3/2} \times [L(v) / 3* \log n ]^{3/2}
= [L(v) / M]^{3/2}
\]

\[
E \left[ \#\text{answers to } Q \text{ reported by all servers} \right] \leq \sum_v [L(v) / M]^{3/2} = p \times [L / M]^{3/2} = [L / L_{\text{lower}}]^{3/2}
\]

\( Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \)

size\( (R) + \text{size}(S) + \text{size}(T) = M \)
Outline

• The MPC Model

• Communication Cost for Triangles

• Communication Cost for CQ’s

• Discussion
General Formula

Consider an arbitrary conjunctive query (CQ):

\[ Q(x_1, \ldots, x_k) = S_1(x_1) \land \cdots \land S_{\ell}(x_{\ell}) \]

\[ \text{size}(S_1) = M_1, \text{size}(S_2) = M_2, \ldots, \text{size}(S_{\ell}) = M_{\ell} \]

We will:

• Give a lower bound \( L_{\text{lower}} \) formula
• Give an algorithm with load \( L_{\text{algo}} \)
• Prove that \( L_{\text{lower}} = L_{\text{algo}} \)
Cartesian Product

\[ Q(x, y) = S_1(x) \times S_2(y) \]

\[ \text{size}(S_1) = M_1, \text{size}(S_2) = M_2 \]
Cartesian Product

Algorithm: factorize \( p = p_1 \times p_2 \)
- Send \( S_1(x) \) to all servers \((h(x), \ast)\)
  - Send \( S_2(y) \) to all servers \((\ast, h(y))\)
- Load = \( \frac{M_1}{p_1} + \frac{M_2}{p_2} \);
  - Minimized when \( \frac{M_1}{p_1} = \frac{M_2}{p_2} = \left(\frac{M_1M_2}{p}\right)^{\frac{1}{2}} \)

\[ L_{\text{algo}} = 2 \left( \frac{M_1 \cdot M_2}{p} \right)^{\frac{1}{2}} \]
Q(x, y) = S_1(x) \times S_2(y)

Cartesian Product

Algorithm: factorize \( p = p_1 \times p_2 \)
- Send \( S_1(x) \) to all servers \((h(x), \ast)\)
  Send \( S_2(y) \) to all servers \((\ast, h(y))\)
- Load = \( M_1 / p_1 + M_2 / p_2 \);
  Minimized when \( M_1 / p_1 = M_2 / p_2 = (M_1M_2/p)^{\frac{1}{2}} \)

Lower bound:
- Let \( L(v) = L_1(v) + L_2(v) \) = load at server \( v \)
- The \( p \) servers must report all answers to \( Q \):

\[
M_1M_2 \leq \sum_v L_1(v) * L_2(v) \leq \frac{1}{2} \sum_v (L(v))^2
\]

\[
\leq \frac{1}{2} p \max_v (L(v))^2
\]

\[
L_{\text{algo}} = 2 \left( \frac{M_1 \cdot M_2}{p} \right)^{\frac{1}{2}}
\]

\[
L_{\text{lower}} = 2 \left( \frac{M_1 \cdot M_2}{p} \right)^{\frac{1}{2}}
\]
A Simple Observation

**Definition** An *edge packing* for $Q$ is a set of atoms with no common variables:

$$U = \{S_{j_1}(x_{j_1}), S_{j_2}(x_{j_2}), \ldots, S_{|U|}(x_{|U|})\} \quad \forall i, k : x_{j_i} \cap x_{j_k} = \emptyset$$

Assume that the three input relations $S_1, S_2, \ldots$ are stored on disjoint servers

**Claim** Any one-round algorithm for $Q$ must also compute the cartesian product:

$$Q'(x_{j_1}, \ldots, x_{|U|}) = S_{j_1}(x_{j_1}) \times S_{j_2}(x_{j_2}) \times \ldots \times S_{|U|}(x_{|U|})$$

**Proof** Because the algorithm doesn’t know the other $S_j$’s.

$$L_{\text{lower}} = |U| \cdot \left( \frac{M_{j_1} \cdot M_{j_2} \cdots M_{|U|}}{|U|} \right) \frac{1}{p}$$

Proof similar to cartesian product on previous slide

*otherwise, add another factor $|U|$*
$Q(x_1, \ldots, x_k) = S_1(x_1) \Join \ldots \Join S_{\ell}(x_{\ell})$  
size($S_1$) = $M_1$, size($S_2$) = $M_2$, \ldots, size($S_{\ell}$) = $M_{\ell}$

### The Lower Bound for CQ

#### Definition

A *fractional edge packing* are real numbers $u_1, u_2, \ldots, u_{\ell}$ s.t.:

\[
\forall j \in [\ell] : \quad u_j \geq 0
\]

\[
\forall i \in [k] : \quad \sum_{j \in [\ell] : x_i \in \text{vars}(S_j(x_j))} u_j \leq 1
\]

#### Theorem

*Any one-round deterministic algorithm $A$ for $Q$ requires $O(L_{\text{lower}})$ bits of communication per server:*

\[
L_{\text{lower}} = \left( \frac{M_1^{u_1} \cdot M_2^{u_2} \cdot \ldots \cdot M_{\ell}^{u_{\ell}}}{u_1 + u_2 + \cdots + u_{\ell}} \right)^{\frac{1}{p}}
\]

Note that we ignore constant factors (like $|U|$ on the previous slide)

*Beame, Koutris, Suciu, Skew in Parallel Data Processing, PODS 2014*
Triangle Query Revisited

\[ Q(X,Y,Z) = R(X,Y) \bowtie S(Y,Z) \bowtie T(Z,X) \]

\[ \text{size}(R) = M_1, \ \text{size}(S) = M_2, \ \text{size}(T) = M_3 \]

Packing \( u_1, u_2, u_3 \) | \( L = (M_1 u_1 \times M_2 u_2 \times M_3 u_3 / p)^{1/(u_1+u_2+u_3)} \)
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{2}, \frac{1}{2}, \frac{1}{2} )</td>
<td>( (M_1 M_2 M_3)^{\frac{1}{3}} / p^{\frac{2}{3}} )</td>
</tr>
<tr>
<td>1, 0, 0</td>
<td>( M_1 / p )</td>
</tr>
<tr>
<td>0, 1, 0</td>
<td>( M_2 / p )</td>
</tr>
<tr>
<td>0, 0, 1</td>
<td>( M_3 / p )</td>
</tr>
</tbody>
</table>

\( L_{\text{lower}} = \text{the largest of these four values} \)

When \( M_1 = M_2 = M_3 = M \), then the largest value is the first: \( M / p^{\frac{2}{3}} \)
An Algorithm for CQ

[Afrati&Ullman’2010]
Factorize \( p = p_1 \times p_2 \times \ldots \times p_k \); a server is \((v_1, \ldots, v_k) \in [p_1] \times \ldots \times [p_k]\)

- Send \( S_1(x_{11}, x_{12}, \ldots) \) to servers with coordinates \( h(x_{11}), h(x_{12}), \ldots \)
- Send \( S_2(x_{21}, x_{22}, \ldots) \) to servers with coordinates \( h(x_{21}), h(x_{22}), \ldots \)
- \( \ldots \)

- Compute the query locally at each server

- Load is \( O(L_{\text{algo}}) \), where:

\[
L_{\text{algo}} = \max_{j \in [\ell]} \frac{M_j}{\prod_{i \in [k]: x_i \in \text{vars}(S_j(x_j))} p_i}
\]

[A&U] use \textit{sum} instead of \textit{max}. With \textit{max} we can compute the optimal \( p_1, p_2, \ldots, p_k \).
$Q(x_1, \ldots, x_k) = S_1(x_1) \otimes \ldots \otimes S_\ell(x_\ell)$  \quad \text{size} (S_1) = M_1, \text{size} (S_2) = M_2, \ldots, \text{size} (S_\ell) = M_\ell$

$L_{\text{lower}} = L_{\text{algo}}$

$L_{\text{algo}} = \max_j \frac{M_j}{\prod_{i: x_i \in \text{vars} (S_j)} p_i}$

$L_{\text{lower}} = \left( \frac{\prod_j M_j^{u_j}}{p} \right)^{\frac{1}{\sum_j u_j}}$

Apply log in base $p$. Denote $\mu_j = \log_p M_j$, $e_i = \log_p p_i$
\[ Q(x_1, \ldots, x_k) = S_1(x_1) \otimes \ldots \otimes S_\ell(x_\ell) \quad \text{size}(S_1) = M_1, \text{size}(S_2) = M_2, \ldots, \text{size}(S_\ell) = M_\ell \]

\[ L_{\text{lower}} = L_{\text{algo}} \]

\[ L_{\text{algo}} = \max_j \frac{M_j}{\prod_{i: x_i \in \text{vars}(S_j)} p_i} \]

\[ L_{\text{lower}} = \left( \frac{\prod_j M_j^{u_j}}{p} \right)^{\frac{1}{\sum_j u_j}} \]

Apply log in base \( p \). Denote \( \mu_j = \log_p M_j, \quad e_i = \log_p p_i \)

\[
\begin{align*}
\text{minimize } & \lambda \\
\sum_i e_i & \leq 1 \\
\forall j : & \lambda + \sum_{j: x_i \in \text{vars}(S_j)} e_i \geq \mu_j
\end{align*}
\]

\[
\begin{align*}
\text{maximize } & \frac{1}{\sum_j u_j} \left( \sum_j u_j \mu_j - 1 \right) \\
\forall i : & \sum_{j: x_i \in \text{vars}(S_j)} u_j \leq 1
\end{align*}
\]
\[ Q(x_1, \ldots, x_k) = S_1(x_1) \times \ldots \times S_\ell(x_\ell) \]

size(\(S_1\)) = \(M_1\), size(\(S_2\)) = \(M_2\), \ldots, size(\(S_\ell\)) = \(M_\ell\)

**Lower Bound (\(L_{\text{lower}}\))**

\[ L_{\text{lower}} = \left( \frac{\prod_j M_j^{u_j}}{p} \right)^{\frac{1}{\sum_j u_j}} \]

**Algorithmic Bound (\(L_{\text{algo}}\))**

\[ L_{\text{algo}} = \max_j \frac{M_j}{\prod_{i : x_i \in \text{vars}(S_j)} p_i} \]

**Primal/Dual Equivalence**

Apply log base \(p\). Denote \(\mu_j = \log_p M_j\), \(e_i = \log_p p_i\)

- Minimize \(\lambda\)
  \[ \sum_i e_i \leq 1 \]
  \[ \forall j : \lambda + \sum_{i : x_i \in \text{vars}(S_j)} e_i \geq \mu_j \]

- Maximize \(\sum_j f_j \mu_j - f_0\)
  \[ \sum_j f_j \leq 1 \]
  \[ \forall i : \sum_{j : x_i \in \text{vars}(S_j)} f_j \leq f_0 \]

Primal/Dual

\[ u_j = \frac{f_j}{f_0} \quad u_0 = \frac{1}{f_0} \]

at optimality: \(u_0 = \sum_j u_j\)
Outline

- The MPC Model
- Communication Cost for Triangles
- Communication Cost for CQ’s
- Discussion
Discussion

Communication costs for multiple rounds
• Lower/upper bounds in [Beame’13] connect the number of rounds to log of the query diameter
• Weaker communication model: algorithm sends only tuples, not arbitrary bits

Communication cost for skewed data
• Lower/upper bounds in [Beame’14] have a gap of poly log p

Communication costs beyond full CQ’s
• Open!